1. 自连接

1.1. 需求

现有这么一批数据，现要求出：

```
每个用户截止到每月为止的最大单月访问次数和累计到该月的总访问次数
A    1    20  20    20
A    2    10  30    20
A    3    30  60    30
A    4    50  110   50
.....
B    1    2   2
B    2    10  12
B    3    20  32
B    4    50  82
```

三个字段的意思：
用户名，月份，访问次数

```
A,2015-01,5
A,2015-01,15
B,2015-01,5
A,2015-01,8
B,2015-01,25
A,2015-01,5
A,2015-02,4
A,2015-02,6
B,2015-02,10
B,2015-02,5
A,2015-03,16
A,2015-03,22
B,2015-03,23
B,2015-03,10
B,2015-03,11
```

```
create table exercise_pv_temp as select name, month, sum(pv) as pv from
exercise_pv group by name, month;
```

结果：

```
A       2015-02 10
B       2015-01 30
A       2015-03 38
B       2015-02 15
A       2015-01 33
B       2015-03 44
```

编写SQL求出一下结果：
最后结果展示:

```
用户      月份    当月访问次数   最大访问次数   总访问次数
A       2015-01     33      33      33
A       2015-02     10      33      43
A       2015-03     38      38      81
B       2015-01     30      30      30
B       2015-02     15      30      45
B       2015-03     44      44      89
```

假设有这种格式的数据，你肯定会写SQL

```
A       2015-01     33  A   2015-01 33
A       2015-02     10  A   2015-01 33
A       2015-03     38  A   2015-01 33

A       2015-01     33  A   2015-02 10
A       2015-02     10  A   2015-02 10
A       2015-03     38  A   2015-02 10

A       2015-01     33  A   2015-03 38
A       2015-02     10  A   2015-03 38
A       2015-03     38  A   2015-03 38
```

```
aname  amonth   apv   bname  bmonth  bpv
```

```
SQL:
select bname, bmonth, bpv, sum(apv) as sumpv,  max(apv) as maxpv from table
group by bname, bmonth, bpv where amonth <= bmonth;
```

```
笛卡尔积
```

```
select a.name as aname, a.month as amonth, a.pv as apv,
b.name as bname, b.month as bmonth, b.pv as bpv from
exercise_pv a join exercise_pv b
on a.name  = b.name;
```

```
最终的SQL:
select aa.bname, aa.bmonth, aa.bpv, sum(aa.apv) as sumpv,  max(aa.apv) as maxpv
from (
select a.name as aname, a.month as amonth, a.pv as apv,
b.name as bname, b.month as bmonth, b.pv as bpv from
exercise_pv_temp a join exercise_pv_temp b
on a.name = b.name
) aa
where aa.amonth <= aa.bmonth
group by aa.bname, aa.bmonth, aa.bpv
order by aa.bname, aa.bmonth;
```

换一种方式：

```
select name,
month,
pv,
sum(pv) over (partition by name order by month asc rows between unbounded
preceding and current row) as spv,
max(pv) over (partition by name order by month asc rows between unbounded
preceding and current row) as mpv
from exercise_pv_temp;
```

```
第一个要点：sum  max  count   min   avg
```

```
partition by name order by month asc rows between unbounded preceding and
current row
partition by name：严格来说是分区，事实上，你完全可以理解成是分组
order by month asc：  每组数据按照month升序排序
rows between A and B：  到底哪些记录作为一组来计算，添加一个窗口的边界
    A：unbounded preceding
       3 preceding    （当前记录不算，往前数3条）
       current row
    B: current row
       3 following     （当前记录不算，往后数3条）
       unbounded following     到这一组的最后为止
```

```
row between 5 preceding  and  2 following
前5条记录
当前记录
后2条记录
这个窗口的长度是：8
```

建表准备：

```
create database if not exists exercise_db;
use exercise_db;
drop table if exists exercise_pv;
create table exercise_pv(name string, month string, pv int) row format delimited
fields terminated by ",";
load data local inpath "/home/bigdata/exercise_pv.txt" into table exercise_pv;
select * from exercise_pv;
desc exercise_pv;
```

开启本地执行模式：

```
set hive.exec.mode.local.auto=true;
```

1.2. 普通自连接实现

第一步：先做按月汇总

```
select a.name, a.month, sum(a.pv) as pv from exercise_pv a group by a.name,
a.month;
```

得到结果：

```
name    month   pv
A       2015-01 33
A       2015-02 10
A       2015-03 38
B       2015-01 30
B       2015-02 15
B       2015-03 44
```

第二步：设计解题思路
1、为了得到最终的结果数据：

```
用户   月份        当月访问次数   最大访问次数   总访问次数
A       2015-01     33      33      33
A       2015-02     10      33      43
A       2015-03     38      38      81
B       2015-01     30      30      30
B       2015-02     15      30      45
B       2015-03     44      44      89
```

2、如果能得到这样的数据：

```
select a.name as aname, a.month as amonth, a.pv as apv,
b.name as bname, b.month as bmonth, b.pv as bpv
from
(select a.name, a.month, sum(a.pv) as pv from exercise0101 a group by a.name,
a.month) a
join
(select a.name, a.month, sum(a.pv) as pv from exercise0101 a group by a.name,
a.month) b
on a.name = b.name
where a.month >= b.month;
```

结果数据：

```
A       2015-01 33      A       2015-01 33
A       2015-02 10      A       2015-01 33
A       2015-02 10      A       2015-02 10
A       2015-03 38      A       2015-01 33
A       2015-03 38      A       2015-02 10
A       2015-03 38      A       2015-03 38
B       2015-01 30      B       2015-01 30
B       2015-02 15      B       2015-01 30
B       2015-02 15      B       2015-02 15
B       2015-03 44      B       2015-01 30
B       2015-03 44      B       2015-02 15
B       2015-03 44      B       2015-03 44
```

3、那么执行这个SQL就行：

```
select a.aname, a.amonth, a.apv,  max(a.bpv) as maxpv, sum(a.bpv) as sumpv
from
(
select a.name as aname, a.month as amonth, a.pv as apv,
b.name as bname, b.month as bmonth, b.pv as bpv
from
(select a.name, a.month, sum(a.pv) as pv from exercise0101 a group by a.name,
a.month) a
join
(select a.name, a.month, sum(a.pv) as pv from exercise0101 a group by a.name,
a.month) b
on a.name = b.name
where a.month >= b.month
) a
group by a.aname, a.amonth, a.apv;
```

就能得到最终结果

```
A       2015-01 33      33      33
A       2015-02 10      33      43
A       2015-03 38      38      81
B       2015-01 30      30      30
B       2015-02 15      30      45
B       2015-03 44      44      89
```

第三步：优化一下得到最终的结果SQL：

```
select a.name as aname, a.month as amonth, a.pv as apv,
max(b.pv) as maxpv, sum(b.pv) as sumpv
from
(select a.name, a.month, sum(a.pv) as pv  from exercise0101 a  group by a.name,
a.month) a
join
(select a.name, a.month, sum(a.pv) as pv  from exercise0101 a  group by a.name,
a.month) b
on a.name = b.name
where a.month >= b.month
group by a.name, a.month, a.pv;
```

执行上面的SQL语句就可以得到最终的结果
可以使用with语法进行改写优化：

```
with
tt as (select a.name, a.month, sum(a.pv) as pv from exercise0101 a group by
a.name, a.month)
select a.name as aname, a.month as amonth, a.pv as apv,
max(b.pv) as maxpv, sum(b.pv) as sumpv
from tt a join tt b
on a.name = b.name
where a.month >= b.month
group by a.name, a.month, a.pv;
```

1.3. 使用开窗函数实现

```
select
a.name,
a.month,
a.pv,
sum(a.pv) over (partition by a.name order by a.month rows between unbounded
preceding and current row) as sumpv,
max(a.pv) over (partition by a.name order by a.month rows between unbounded
preceding and current row) as maxpv
from
(select b.name as name, b.month as month, sum(b.pv) as pv from exercise_pv b
group by b.name, b.month) a;
```

最终结果：

```
+---------+----------+-------+--------+--------+
| a.name  | a.month  | a.pv  | sumpv  | maxpv  |
+---------+----------+-------+--------+--------+
| B       | 2015-01  | 30    | 30     | 30     |
| B       | 2015-02  | 15    | 45     | 30     |
| B       | 2015-03  | 44    | 89     | 44     |
| A       | 2015-01  | 33    | 33     | 33     |
| A       | 2015-02  | 10    | 43     | 33     |
| A       | 2015-03  | 38    | 81     | 38     |
+---------+----------+-------+--------+--------+
```